Confidential information

Amendments to Super Video Compact Disc SVCD System Specifications Version 1.0 May, 1999

Reference	#001 Withdrawn

Reference	#002 PS header
Date	10/8/99
Clause	V.2.2 PS system_header Table V-2, PS system_header constraints, stream_id (Note) (SVCD version 1.0, page25)
Existing text	specify an entry for every PES present in the PS ...
New text	specify an entry for every audio and video PES present in the PS ...
Justification	It is not needed to specify an entry for private_stream_1

Reference	#003 Table reference error
Date	10/8/99
Clause	IV.3.1 INFO.SVD file
Existing text	System Identification -- This 8 character field is coded as ISO 646 upper case characters, and the string value is defined in table 6-4.
New text	System Identification -- This 8 character field is coded as ISO 646 upper case characters, and the string value is defined in Table IV-4.
Justification	wrong table reference

Reference	#004 Table reference error
Date	10/8/99
Clause	V.6.1 User data structure Table V-30 General structure of User Data (page 41)
Existing text	user_data_group #N : Table III-1
New text	user_data_group #N : see Table V-31
Justification	wrong table reference

For further information, please contact:
Philips Intellectual Property & Standards, Business Support, Building SFF-8, P.O. Box 80002, 5600 JB Eindhoven, The Netherlands, Fax. : +31-40-2732113
Or check our web site: http://www.licensing.philips.com

PHILIPS

Reference	#005 ISO 646 subset
Date	21/9/99
Clause	I.4 Conventions and symbols (page 3)
Existing text	Character String -- Character strings are always given between double quotation marks, as "_____", coded according to ISO 646 if not indicated otherwise.
New text	Character String -- Character strings are always given between double quotation marks, as "_____", coded according to ISO 646 if not indicated otherwise. Fixed length descriptor fields are left-justified and remaining positions to the right are filled with either Space ($20) or Null ($00) characters.
Justification	Clarification of padding

Reference	#006 Bit ordering
Date	21/9/99
Clause	IV.3.1 INFO.SVD file Table IV-6 Segment Play Item Contents byte (page 13)
Existing text	Table starts with bit#0
New text	Change table according to (msb) starting with bit#7
Justification	Clarification to avoid confusion on bit-ordering

Reference	#007 Bit ordering
Date	21/9/99
Clause	IV.3.6 TRACKS.SVD file Table IV-13 Track Contents byte (page 18)
Existing text	Table starts with bit#0
New text	Change table according to (msb) starting with bit#7
Justification	Clarification to avoid confusion on bit-ordering

Reference	#008 List ID's
Date	24/9/99
Clause	VII.4 Command List 1st paragraph, row 4 page(51)
Existing text	As all Lists starts with an List_ID byte, …
New text	As all Lists starts with a Header byte, …
Justification	Type error

Reference	#009 CD-ROM XA restrictions
Date	11/10/99
Clause	III General SVCD Disc format
Existing text	The general SVCD disc structure is based on CD-ROM (ISO/IEC 10149) and CD-ROM XA with the specific additions and restrictions as defined in this chapter.
New text	The general SVCD disc structure is based on CD-ROM (ISO/IEC 10149) and CD-ROM XA Chapter I to III.2.1 plus III.2.7, with the specific additions and restrictions as defined in this chapter.
Justification	Clarification of what parts of CD-ROM XA apply

PHILIPS

Reference	#010 Command List ID
Date	11/10/99
Clause	VII.4 Command List
Existing text	List ID -- See the description of List ID of Play List.
New text	List ID -- See the description of List ID of Play List. The List ID should have the same value as the List ID of the related preceded List.
Justification	Allows more robust check to verify a valid Command List

Reference	#011 Command List $F0
Date	11/10/99
Clause	VII.4.2.2 Player registers
Existing text	Register $F0 reserved
New text	List_Offset – This R/W register contains at Read the value of the List_Offset of the current Command List. This function is the general method to jump to an other List. At Write the Command List shall be aborted and the List defined by the List_Offset value written to the register shall be played.
Justification	Improves the PSD functionality, and was part of version 0.9

Reference	#012 Command list calculate_2 function
Date	14/10/99
Clause	VII.4.3.5 opcode function definitions (page 59)
Existing text	calculate_2 function -- If cond2 TRUE then …
New text	calculate_2 function -- If cond1 TRUE then …
Justification	type error

Reference	#013 Command list Calculate opcode examples
Date	14/10/99
Clause	VII.4.3.5 opcode function definitions Table VII-18 calculate opcode(xxx) syntax (page 59)
Existing text	V[j]=V[k] + V[l] V[j]=V[k] - V[l] V[j]=V[k] + V[l] V[j]=V[k] + V[l] V[j]=V[k] + V[l] V[j]=V[k] + V[l] V[j]=V[k] + V[l] V[j]=V[k] + V[l]
New text	V[j]=V[k] + V[l] V[j]=V[k] - V[l] V[j]=V[k] * V[l] V[j]=V[k] / V[l] V[j]=V[k] % V[l] V[j]=V[k] & V[l] V[j]=V[k] | V[l] V[j]=V[k] ^ V[l]
Justification	type error

Reference	#014 Command list loop functions
Date	14/10/99
Clause	VII.4.3.5 opcode function definitions opcode loop_x functions (page 60)
Existing text	Loop_1 function -- Decrement variable V[i], defined by byte #2, and jump to the command with index = idx defined by byte #4 and #5. loop_2 function -- Decrement variables V[i], defined by byte #2 and V[j], defined by byte #3, and jump to the command with index = idx defined by byte #4 and #5.
New text	loop_1 function -- if V[i]>0 then decrement variable V[i], defined by byte #2, and jump to the command with index = idx defined by bytes #4 and #5. loop_2 function -- if V[i]>V[j] then decrement variable V[i], defined by byte #2, and jump to the command with index = idx defined by bytes #4 and #5. loop_3 function -- if V[i]>0 then decrement variables V[i], defined by byte #2 and V[j], defined by byte #3, and jump to the command with index = idx defined by bytes #4 and #5.
Justification	type errors and missing text

Reference	#015 loop3 definition
Date	14/10/99
Clause	VII.4.3.2 VM commands Table VII-15 VM opcode and operand definition table (page 58)
Existing text	loop1: while V[i]>0 {--V[i]; jump to command #idx} loop2: while V[i]>V[j] {--V[i]; jump to command #idx} loop3: while V[i] {--V[i]; --V[j]); jump to command #idx}
New text	loop1: if V[i]>0 { --V[i]; jump to command #idx } loop2: if V[i]>V[j] { --V[i]; jump to command #idx} loop3: if V[i]>0 { --V[i]; --V[j]; jump to command #idx}
Justification	type error

Reference	#016 Command list register default values
Date	18/10/99
Clause	VII.4.2.2 Player registers (page 54)
Existing text	The player shall set the defined player registers reflecting the current player status to be read by the VM. The VM can control the player …
New text	The player shall set the defined player registers reflecting the current player status to be read by the VM. At startup of the disc the default player values should be used. Some players allow the user to change the default values. The VM can control the player …
Justification	Clarification

Reference	#017 Command list register user_input
Date	18/10/99
Clause	VII.4.2.2 Player registers user_Input (page 55)
Existing text	User_input – This R/W register contains at … If the user input is disabled then the user has no playback control.
New text	User_input – This R/W register contains at … If the user input is disabled then the user has no playback control. At start up of the disc the register should be set to 255 (User_input enabled).
Justification	Clarification

Reference	#018 Command list register PlayList Play Item wait time
Date	18/10/99
Clause	VII.4.2.2 Player registers PlayList Play Item wait time (page 56)
Existing text	PlayList Play Item wait time -- If current PSD list is a Play List, then this R/W register contains at Read the Play Item Wait Time value as defined by the current Play List, and Write this value can be set to a new value, overruling the old value. See clause VII.2 for definition of Play Item Wait Time.
New text	PlayList Play Item wait time -- If current PSD list is a Play List, then this R/W register contains at Read the Play Item Wait Time value as defined by the current Play List, and Write this value can be set to a new value, overruling the old value. If the current PSD list is not a Play List then the register should be set to -1. See clause VII.2 for definition of Play Item Wait Time.
Justification	Clarification

Reference	#019 Command list register Selection List wait time for timeout
Date	18/10/99
Clause	VII.4.2.2 Player registers Selection List wait time for timeout (page 56)
Existing text	Selection List wait time for timeout -- If current PSD list is a Selection List, then this R/W register contains at Read the Wait Time For Timeout value as defined by the current Play List, and Write this value can be set to a new value, overruling the old value. See clause VII.3 for definition of Wait Time For Timeout.
New text	Selection List wait time for timeout -- If current PSD list is a Selection List, then this R/W register contains at Read the Wait Time For Timeout value as defined by the current Play List, and Write this value can be set to a new value, overruling the old value. If the current PSD list is not a Selection List then the register should be set to -1. See clause VII.3 for definition of Wait Time For Timeout.
Justification	Clarification

Reference	#020 Command list register Shuffle
Date	18/10/99
Clause	VII.4.2.2 Player registers Shuffle (page 56)
Existing text	Shuffle -- This R/W register contains at Read the next value from the Shuffle sequence, and at Write a new random Shuffle sequence 1..N is generated when N is set to the register, and N>0 and N<256. A Shuffle sequence shall contain all values 1..N only once in random order. At end of the sequence the register shall be set to 0.
New text	Shuffle -- This R/W register contains at Read the next value from the Shuffle sequence, and at Write a new random Shuffle sequence 1..N is generated when N is set to the register, and 0<256. A Shuffle sequence shall contain all values 1..N only once in random order. At end of the sequence the register shall be set to 0. This also disables the sequence and further read will return 0. To start a new sequence a value N has to be written into the register.
Justification	Clarification

Reference	#021 Disabled selection areas
Date	3/11/99
Clause	VII.3.1 Selection Areas Fields First paragraph (page 50)
Existing text	This part of the Selection List is only valid if the Selection Area Extension flag (bit#0) is set to %1
New text	This part of the Selection List is only valid if the selection_area_flag (bit#0) is set to %1. Disabled Selection Area fields (the corresponding offset = $FFFF) should be set to zero (0,0,0,0).
Justification	Type error and missing text.

Reference	#022 Extension_flag
Date	3/11/99
Clause	VII.3 Selection List Table VII-5 extension_flag (page 48)
Existing text	if extension_flag ==’1’ {
New text	if (selection_area_flag ==’1’) {
Justification	Type error

Reference	#023 Extension bit
Date	3/11/99
Clause	VII.3 Selection List Table VII-6 Note (page 48)
Existing text	Note: If the Extension bit is set then ...
New text	Note: If the selection_area_flag is set to %1 then ...
Justification	Type error

Reference	#024 List ID
Date	3/11/99
Clause	VII.2 Play list Table VII-2 (page 46)
Existing text	Contents for bit 0..14 is “$0001 .. $7FFFF”
New text	Contents for bit 0..14 should be “$0001 .. $7FFF”
Justification	Type error

Reference	#025 Selection list selection_offset()
Date	20/1/00
Clause	VII.3 Selection List Table VII-5 Selection List Structure (page 48)

Existing text	Play Item Number
	for (i=BSN; j<BSN+NOS; i++) {
	if (Default list Offset <$FFFD) Selection_offset(i)
	if (Default list Offset == $FFFD or Default list Offset == $FFFE)
	Multi_default_selection_offset(i) }

New text	Play Item Number
	for (i=BSN; j<BSN+NOS; i++) {
	if (Default list Offset == $FFFD or Default list Offset == $FFFE)
	Multi_default_selection_offset(i)
	else Selection_offset(i) }

Justification	Syntax error, the loop now specify the case when Default List Offset = $FFFF

Reference	#026 Set Random Command
Date	14/2/00
Clause	VII.4.3.5 Opcode function definitions (page 59)

Existing text	if cond1 { V[j] = random value;
	0<=V[j]<=dddd };
	if (dddd == $0000) {randomize/change seed}

New text	if cond1 { if dddd>0 { V[j] = random_value(); 0<=V[j]<=dddd} else randomize()}
Justification	Clarification for values of dddd<0

Reference	#027 Set Random Command
Date	14/2/00
Clause	VII.4.3.2 VM commands Table VII-15 VM opcode and operand definition table (page 58)
Existing text	random function -- The variable V[j] indexed by byte #3 is set to a random value between 0 (zero) and dddd defined by byte #4 and #5. If dddd=0 then randomize the random generator
New text	random function -- If dddd>0, where dddd is defined by byte #4 and #5, then the variable V[j] indexed by byte #3 is set to a random value between 0 (zero). If dddd<=0 then randomize (change seed of) the random generator and V[j] shall not be affected for any value of j.
Justification	Clarification for values of dddd<0

Reference	#028 Endlist Next_disc number
Date	3/2/00
Clause	VII.5 End List (page 60)
Existing text	Next_disc -- This one byte entry identifies next disc #nn to play of current Album. If Next_disc is equal to $00 then this indicates end of playing.
New text	Next_disc -- This one byte entry identifies next disc to play of current Album, where the ordinal number $nn is equal to the Album Sequence Number in INFO.SVD incremented by one. If Next_disc is equal to $00 then this indicates end of playing.
Justification	Definition error. The Album Sequence Number in INFO.SVD starts from 0, so it was impossible to go back to the first disc of a set.

Reference	#029 Command list opcode conditions
Date	6/3/00
Clause	VII.4.3.4 opcode conditions (page 59)
Existing text	… The cond_1 test compares a variable V[i] compared to 0, and the cond_2 test compares the relation between two variables V[i] and V[j] as defined below.
New text	… The cond_1 test compares a variable V[i] compared to 0, and the cond_2 test compares the relation between two variables V[i] and V[j] or between V[i] and j, where j can have the byte value 0..255, as defined below.
Justification	Clarification

Reference	#030 Fill function command
Date	25/4/00
Clause	VII.4.3.5 opcode function definitions (page 59)
Existing text	fill function -- Fill N variables starting from V[j] with dddd defined by byte #4 and #5, where N is equal the value of V[i] indexed by byte #2
New text	fill function -- Fill i variables starting from V[j] with dddd defined by byte #4 and #5, where i is indexed by byte #2, and j by byte #3.
Justification	Type error

Reference	#031 OGT CLUT data
Date	26/6/00
Clause	V.5.4.2 OGT CLUT data Table V-22 CLUT_data Syntax (page 36)
Existing text	CLUT_data() {if (CLUT_type == '0010') {for (i=0;i<4;i++) { Y_value Cb_value Cr_value mix_ratio } } }
New text	CLUT_data() {if (CLUT_type == '0010') {for (i=0;i<4;i++) { Y_value Cr_value Cb_value mix_ratio} } }
Justification	Type error. The Cb and Cr values were in the wrong order, not compliant with implementations and other standards.